On a Hashing-Based Enhancement of Source Separation Algorithms over Finite Fields with Network Coding Perspectives

Irina-Delia Nemoianu, Claudio Greco, *Member, IEEE*,
Marco Cagnazzo, *Senior Member, IEEE,* Béatrice Pesquet-Popescu, *Fellow, IEEE*

***Abstract*— Blind Source Separation (BSS) deals with the recovery of source signals from a set of observed mixtures, when little or no knowledge of the mixing process is available. BSS can find an application in the context of network coding, where relaying linear combinations of packets maximizes the throughput and increases the loss immunity. By relieving the nodes from the need to send the combination coefficients, the overhead cost is largely reduced. However, the scaling ambiguity of the technique and the quasi-uniformity of compressed media sources makes it unfit, at its present state, for multimedia transmission. In order to open new practical applications for BSS in the context of multimedia transmission, we have recently proposed to use a non-linear encoding to increase the discriminating power of the classical entropy-based separation methods. Here, we propose to append to each source a non-linear message digest, which offers an overhead smaller than a per-symbol encoding and that can be more easily tuned. Our results prove that our algorithm is able to provide high decoding rates for different media types such as image, audio, and video, when the transmitted messages are less than 1.5 kilobytes, which is typically the case in a realistic transmission scenario.**

***Index Terms*—Blind Source Separation; Channel Coding; Galois Fields; Independent Component Analysis; Network Coding; Multimedia Networking.**

I. Introduction

In *Network Coding* (NC) [1], instead of merely relaying packets, the intermediate nodes of a network send linear combinations of the packets they have previously received, with random coefficients taken from a finite field [2–5]. NC, used as an alternative to traditional routing, has proved beneficial to real-time streaming applications, both in terms of maximization of the throughput and in terms of reduction of the effects of losses [6–12]. However, in Practical Network Coding (PNC) [5] approaches, the random coefficients must be included in the packet as headers, incurring a fixed overhead that can be prohibitive if the maximum packet size is small.

A few work exist that try to reduce this overhead. Jafari *et al.* [13] proposed to exploit the sparsity of the coding vectors to compress the coding coefficients, which can indeed reduce the overhead, but only if the number of packets in each combination is much smaller than the size of the generation. Li and Ramamoorthy [14] introduce two new packet formats that allow to use erasure decoding and list decoding. These new solutions have a smaller overhead than classical approaches, at least when the number of sources is large enough. Finally, another approach, proposed by Thomos and Frossard[15], consists in generating the coding vectors in such way that the coding operations can be described by using just one symbol per packet. The encoding vectors are the rows of a (modified) Vandermonde matrix. This approach also can reduce the overhead, but may increase the probability of generating a singular global encoding matrix, rendering the received packets non-decodable.

Another limitation of PNC is that it implicitly assumes that all the nodes of the network have previously agreed on a complete ordering of the sources, that is, a correspondence between a source and the column in the global encoding vectors where its coefficient will be stored. In networks with high churn, the consensus problem over the source ordering may be non-trivial.

We argue that the fixed overhead incurred by practical network coding schemes may be ill-suited to deal with content in which different segments may have different impacts on the users' satisfaction, such as the case of multimedia. Our goal, is thus to design a network coding technique in which decoding probability can be traded off with the transmission overhead, so that the two can be integrated in a transmission system providing unequal loss protection to different parts of the content, *e.g.*, based on their impact on the total distortion. In order to do so, we propose to avoid inclusion of the coding coefficients altogether (which solves the consensus problem), and to replace it with a hashing of the packet content. At the decoder side, in order to decode the packets, we formulate the unknown encoding matrix inversion as a *Blind Source Separation* (BSS) [16] problem.

Generally speaking, BSS consists in recovering a set of N source signals $\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_N$, organized in a matrix

$$\mathbf{S} = \begin{pmatrix} \mathbf{s}_1 \\ \vdots \\ \mathbf{s}_N \end{pmatrix}$$

from a set of mixed signals $\mathbf{Y} = f(\mathbf{S})$ (also referred to as

Authors are with the Institut Mines-Télécom, Télécom ParisTech, CNRS LTCI, TSI Department, 46 rue Barrault, 75634 Paris, France, (e-mail: {nemoianu,greco,cagnazzo,pesquet}@telecom-paristech.fr). C. Greco is also affiliated with INRIA Rocquencourt, HIPERCOM Project Team, Domaine de Voluceau, BP 105, 78153 Le Chesnay, France (e-mail: claudio.greco@inria.fr) and LSS-Supélec, CNRS, Division Télécoms et Réseaux, 3 rue Joliot-Curie, 91192 Gif-sur-Yvette, France (e-mail: claudio.greco@lss.supelec.fr).

observations) without knowing the sources themselves nor the parameters of the mixing process. This is a subject that has been intensively investigated in the last three decades, due to its numerous potential applications in fields such as neural networks, speech recognition, sensor signal processing, image restoration, and biomedical signal processing [16, 17].

In the context of BSS, the sources are described as random vectors over the real or complex field, while the mixing process consists in a *linear* combination with coefficients taken from the same field as the source signals:

$$\mathbf{Y} = \mathbf{AS}.$$

In this case, the separation process is reduced to finding the combination matrix $\mathbf{A}$ or, equivalently, its inverse $\mathbf{A}^{-1} = \mathbf{W}$.

The *Independent Component Analysis* (ICA) [18, 19] approach tries to solve the BSS problem by relying on the assumption that the sources are identically distributed and loosely correlated, or statistically independent, with each other. Given a set of observations, an ICA algorithm will return a set of signals that maximizes the regularity of each signal and minimizes the similarity among signals. Several mathematical tools can be used to implement the concept of regularity, such as the entropy or the Kullback-Leibler divergence. In order to obtain good results from ICA, the sources must generally be non-Gaussian. In fact, most algorithms assume –either directly or indirectly– non-Gaussianity as a measure of regularity and rely on the fact that, being a linear combination of several *i.i.d* random variables, the observations will be "more Gaussian" (*e.g.*, in terms of kurtosis), motivated by the Central Limit Theorem, and have a higher entropy than the original sources.

Whatever *contrast function* is used to discriminate between sources and mixtures, one should note that the original sources can only be retrieved up to some ambiguities. Namely, there will be a permutation ambiguity (*i.e.*, the algorithm will not be able to tell which reconstructed source is which) and scaling ambiguity (*i.e.*, the reconstructed sources will be identified up to a scaling factor). For linear mixing, the ambiguities in the reconstructed sources $\widehat{\mathbf{S}}$ can be expressed in the form:

$$\widehat{\mathbf{S}} = \mathbf{\Sigma} \cdot \mathbf{\Pi} \cdot \mathbf{S}.$$

where $\mathbf{\Sigma}$ is a scaling matrix, *i.e.*, a diagonal matrix of scaling factors, and $\mathbf{\Pi}$ is a permutation matrix.

The problem of ICA has been recently extended to the case of finite fields [20], which presents several additional challenges for ICA, due to the nature of the operations defined over a finite field. In particular, the Central Limit Theorem, which is used in real-valued ICA, does not hold true in a finite field. However, entropy minimization can nevertheless be used to separate the sources, as the entropy of any linear combination of statistically independent random variables over a finite field of q elements (denoted $\mathbb{F}_q$) is not less than the entropy of any of the components (as long as none of the components is uniform). Separation is therefore possible by finding the inverse linear transformation that minimizes the marginal entropy of the resulting combinations. Since the operations take place in a finite field, an exhaustive approach is possible, *i.e.*, to try any possible linear combinations of observations until we find the one that has the lowest entropy [20].

The scope of this article is focused on maximizing the discriminating power of the contrast function in order to increase the decoding probability at the receiver. We shall therefore compare ourself to the *Ascending Minimization of Entropies for ICA* (AMERICA) method [20], originally proposed for $\mathbb{F}_2$. This method extracts a single source, then removes the contribution of this source from the mixtures and repeats this process N times, after which it has found all N sources, restricting the search space to vectors linearly independent from the ones recovered so far. This method is guaranteed to always find a set of demixing coefficients that yield reconstructed sources with minimal entropy. Our technique will also follow a similar approach, but the search space will be restricted to vectors that also yield admissible sources, *i.e.*, sources that carry a valid digest.

Other separation algorithms for finite fields have been proposed to reduce the search space –and therefore the execution time– of blind source separation algorithms, at the expenses of the accuracy [21, 22]. One such technique has been proposed for finite fields of prime order only, but can be easily extended to the general case [20]. At each iteration, the algorithm finds a couple of observation vectors $\mathbf{y}_i$ and $\mathbf{y}_j$ and a scalar α in the finite field such that $H(\mathbf{y}_i + \alpha\mathbf{y}_j) < H(\mathbf{y}_i)$ and replaces $\mathbf{y}_i$ with $\mathbf{y}_i + \alpha\mathbf{y}_j$. When no substitution that reduces the entropy can be found, the algorithm terminates, and the final value of the $\mathbf{y}_i$ will be the reconstruction of the original sources. This algorithm is significantly faster than an exhaustive search, but is prone to get stuck in local minima. Other methods to speed up the execution have been proposed, *e.g.*, approximating the entropy with $-p_{\max}\log(p_{\max})$, where $p_{\max}$ is the probability of the most probable element [21, 22].

Even though we mostly focus on low-overhead network coding, it should be noticed that other applications for an efficient blind source separation technique in finite field exist, *e.g.*, in the context of eavesdropping over MIMO multi-user digital communications systems [23].

In this article we improve the results of the separation method by increasing the discriminating power of the algorithm without adding constraints on the distribution of the sources, *i.e.*, not relying on the assumption that none of the original sources is uniform or close to uniform. The rationale is that many of the sources in today's multimedia applications do indeed have a distribution close to the uniform (*e.g.*, compressed images, sound, and video, considered at a suitable level, *i.e.*, bit or byte), so the traditional entropy-based methods would fail in this case. We propose to append to the sources a non-linear message digest, generated by a hashing function, which, as we will show, increases the separability of the ICA method, lending it to more practical application with short sources and distributions closer to uniform.

Although our technique might not always be able to guarantee that $100\,\%$ of transmitted packets are correctly decoded, it does provide a flexible trade-off between transmission rate and decoding probability. This allows for the design of a system where traditional PNC and our technique coexist, providing an unequal error protection to different parts of a media stream, thus enabling a rate-distortion control mechanism in

the transmission scheme.

The rest of this article is organized as follows: in Sec. II, we introduce our proposed approach for generating a variable-length digest of the sources to assist the separation process, with particular attention to the design of the hashing function. Then in Sec. III we validate our technique with experimental results and a comparison with a state-of-the-art exhaustive entropy-based source separation algorithm, for different number of sources, defined in finite fields of different size, and for different types of source distributions. Finally, in Sec. IV, we draw our conclusions and outline some future work.

II. Proposed approach

In this section, we describe our proposed method to separate a number of linearly combined (*mixed*) independent sources defined in a finite field. In this context, the sources are blocks of data, *e.g.*, from a media bitstream, considered as vectors over a finite fields.

A. Prior Work

In general, source separation methods work by applying a function, referred to as *contrast function*, that yields measurably different results when applied to an original source as opposed to a combination of sources, or mixture. For instance, in entropy based methods, the original sources are assumed to have lower entropy than the mixtures, therefore, entropy minimization can be used as a contrast criterion.

In our previous work [24] we have shown that the ability of a contrast function to discriminate sources from mixtures can be augmented if a pre-processing step is taken to introduce an easily detectable feature, or *signature*, in the original sources. In order to reduce the number of false positives, *i.e.*, mixtures carrying a valid signature, the feature must present a low probability of randomly appearing in the mixing process.

In the context of source signals in a finite field, we proposed to use channel encoding of the original sources as signature feature. Since in a linear code –by definition– *any* combination of the codewords is still a codeword, it would be impossible to use a linear code to discriminate between original sources and mixtures. We therefore focus only on non-linear codes. In particular, we opted for a odd-parity bit-code, *i.e.*, adding a bit to each symbol such that the total number of one-bits in each codeword is odd, which is a simple yet effective signature. The odd-parity code is clearly non-linear, as zero is not a codeword.

Our experimental results showed that the entropy based methods benefit from error detecting coding by applying the estimation of the entropy only to solutions that are admissible, in the sense that the reconstructed sources are codewords. In this way, several solutions are eliminated that, even if they present low entropy and could be mistakenly identified as sources by the reference technique, cannot actually correspond to original sources as they are not part of the code.

Further investigation, presented here for the first time, also shows that similar results can be achieved with codes that can detect a higher number of bit errors. In particular, we have tested Hamming codes and Reed-Solomon codes, rendered non-linear by complementing the redundant part. Some of these new results for the $(255, 251)$ Reed-Solomon code for different sizes of the finite field and different lengths of the two sources are given in Table I. However, the fixed structure of these codes implies a fixed –and non-negligible– amount of overhead.

	$T = 256$	$T = 1024$	$T = 4096$
$\mathbb{F}_8$	0.01 (0.03)	0.03 (0.17)	0.04 (0.17)
$\mathbb{F}_{16}$	0.01 (0.06)	0.04 (0.17)	0.04 (0.19)
$\mathbb{F}_{32}$	0.02 (0.09)	0.04 (0.19)	0.04 (0.19)
$\mathbb{F}_{64}$	0.02 (0.09)	0.04 (0.19)	0.04 (0.20)
$\mathbb{F}_{128}$	0.02 (0.11)	0.04 (0.20)	0.04 (0.20)
$\mathbb{F}_{256}$	0.03 (0.14)	0.04 (0.20)	0.04 (0.20)

Table I
Increase in the average number of identified sources when the $(255, 251)$ Reed-Solomon code, rendered non-linear by complementing the redundant part, is used to augment the discriminating power w.r.t. to a purely entropy-based method, for 2 sources and probability of each bit to be equal to one $55\,\%$. Numbers in parentheses refer to the identification when the scaling ambiguity is not tolerated.

B. Novel Contributions

In this work, we propose a more flexible framework, able to control the amount of overhead introduced with the signature feature, thus allowing the user to strike the trade-off best suited for the specific content.

In order to do so, rather than encoding each of the symbols of the sources with a pre-defined error-detecting code, we apply a *hashing function* to the whole sources to generate a variable-length *message digest*. The difference between these approaches can be seen in Fig. 1, where we show the overhead introduced by the feature w.r.t. the original source.

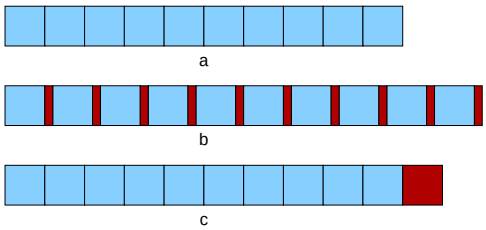


Figure 1. Different approaches to introducing a signature feature. (a) Original symbols of the source. (b) Per-symbol error-detecting encoding. (c) Message digest.

Notice that, in a multi-hop transmission scheme using our technique, only the decoding nodes need to perform the separation. Intermediate nodes of the network perform the same operations in PNC as in our approach[1]. More precisely, both in our approach and in PNC, intermediate nodes just combine received packets using local encoding vectors. As detailed in the following, in the case of PNC the source includes an identity matrix, and the end-receiver will find in those positions the global encoding matrix, which can be inverted to recover the original packets. This approach

[1]The operations required to determine the rank of the reception buffer might slightly change.

requires the inclusion of all coefficient, and thus has a rate of $N \cdot \log_2 q$, where N is the generation size and q is the size of the finite field. In our approach, on the other hand, the source instead of the identity matrix includes a smaller digest, and the decoding nodes will rely on source separation in order to estimate the inverse of the global coding matrix. In summary, the intermediate nodes are completely unaware of either scheme, while source and receiver perform the same task in different ways, in particular while in PNC offers a fixed trade-off between header length and decoding probability, our scheme is flexible in this respect. It is also worth mentioning that, either by mixing different parts of the same content (intra-session NC), or different contents (inter-session NC), or a combination of the two, the final result is always that the end-node receives the product of a generation of source messages and an encoding matrix that has to be inverted. In other words, both transmission scenarios end up with the same separation problem, which our technique can be used to solve.

A hashing function is an algorithm that maps large data sets of variable length into smaller sets. The input of a hashing function is referred to as *message*, whereas its output is referred to as *digest*. These functions are designed so that they are easy to compute, and so that it is unfeasible to generate a message with a given digest, or to modify a message without changing its digest, or to find two different messages having the same digest [25]. Let us denote our hashing function as $\varphi(\cdot) : \mathbb{F}_q^T \mapsto \mathbb{F}_q^D$, where T is the length the message and D the length of the digest, both expressed in number of symbols.

In our context, our hashing function has to be robust w.r.t. linear combinations, rather than a malicious agent or a bit error probability. In other words, the digest of a generic linear combination of sources should not be equal to the same linear combination of their digests, *i.e.*, for a set of N distinct sources $\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_N$, organized in a $N \times T$ matrix $\mathbf{S}$, and a vector $\mathbf{w} = (w_1, \ldots, w_N)$ of combination coefficients:

$$\varphi\left(\mathbf{w}^\top \mathbf{S}\right) \neq \mathbf{w}^\top \Phi(\mathbf{S}),$$

where

$$\Phi(\mathbf{S}) = \begin{pmatrix} \varphi(\mathbf{s}_1) \\ \vdots \\ \varphi(\mathbf{s}_N) \end{pmatrix}$$

Notice that assuming that the N sources are distinct does not lead to a loss of generality: in fact, if we assume that only $N' < N$ source are distinct, we notice that mixing the N sources with a $N \times N$ matrix $\mathbf{A}$ will produce the same observations as mixing the N' distinct sources with an $N' \times N'$ matrix $\mathbf{A}'$. Due to the generality of both N and $\mathbf{A}$ in our discussion, we can therefore always reduce to the case of distinct sources.

If we define a set

$$\mathcal{C}_\varphi = \left\{ \xi \in \mathbb{F}_q^{T'} \mid \xi = (\sigma, \varphi(\sigma)), \forall \sigma \in \mathbb{F}_q^T \right\},$$

with $T' = T + D$, this condition can be equivalently expressed for a set of N distinct vectors $\mathbf{x}_1, \ldots, \mathbf{x}_N$, organized in a $N \times T'$ matrix $\mathbf{X} = (\mathbf{S}\ \Phi(\mathbf{S}))$ as:

$$\mathbf{w}^\top \mathbf{X} \notin \mathcal{C}_\varphi \tag{1}$$

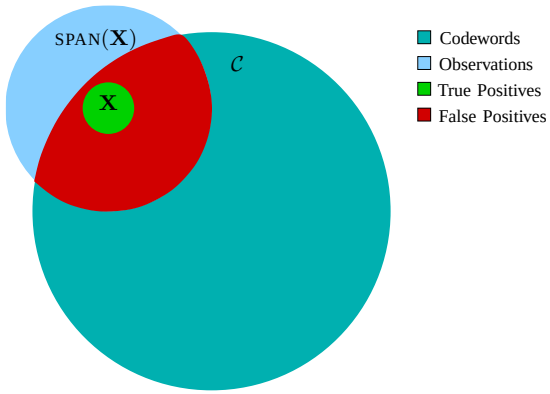


Figure 2. Relation between codewords, observations, real positives and false positives.

Set $\mathcal{C}_\varphi$ is in fact a *code* over $\mathbb{F}_q^{T'}$ and any vector in it is a *codeword*. Our code $\mathcal{C}$ associates to each source of T elements in $\mathbb{F}_q$ the concatenation of the source itself and D additional symbols of digest, generated with a hashing function $\varphi(\cdot)$.

The condition expressed in Eq. (1) allows the digest values to be used to distinguish between the linear combinations and the original sources. However, note that –since the field is finite– it is impossible to design a function φ with $D < T$ for which the non-linearity condition is satisfied for all $\mathbf{w}$ and $\mathbf{S}$. A discriminating hashing function shall therefore present no false negatives, in the sense that an original source always carries a valid digest, but it will also always return some false positives, in the sense that some mixtures will carry a valid digest.

The relative frequency of codewords within the observations of a set of sources $\mathbf{X}$ (*i.e.*, the set of all possible combinations of the vectors $\mathbf{x}_1 \ldots \mathbf{x}_N$) can be expressed as:

$$P(\mathbf{X}) = \frac{\text{Number of observations that are codewords}}{\text{Total number of observations}}$$

Notice that for each matrix $\mathbf{X}$ there will always be at least N observations that are codewords, *i.e.*, the true positives $\mathbf{X}$, therefore minimizing $P(\mathbf{X})$ is equivalent to minimizing the number of false positives.

The value of $P(\mathbf{X})$ can be found by observing that the set of all possible observations given a matrix $\mathbf{X}$ correspond to the linear row span of $\mathbf{X}$, and that an observation is a codeword if it belongs at the same time to the span of $\mathbf{X}$ and to the code $\mathcal{C}_\varphi$ (see Fig. 2).

$$P(\mathbf{X}) = \frac{\|\mathcal{C}_\varphi \cap \text{SPAN}(\mathbf{X})\|}{\|\text{SPAN}(\mathbf{X})\|}.$$

Starting from this definition, first of all we observe that:

$$\mathbf{X} \subseteq \mathcal{C}_\varphi \cap \text{SPAN}(\mathbf{X}) \subseteq \text{SPAN}(\mathbf{X}).$$

The case $\mathcal{C}_\varphi \cap \text{SPAN}(\mathbf{X}) = \text{SPAN}(\mathbf{X})$ is verified when $\text{SPAN}(\mathbf{X}) \subseteq \mathcal{C}_\varphi$, which corresponds by definition to the case of $\mathcal{C}_\varphi$ being a linear code. We observe in this case that $P(\mathbf{X}) = 1$ for all $\mathbf{X}$, consistently with our considerations above. Conversely, when $\mathcal{C}_\varphi \cap \text{SPAN}(\mathbf{X}) = \mathbf{X}$, the code is *perfectly non-linear*, in the sense that there are no linear combinations of vectors in $\mathbf{X}$ that belong to the code other

that the product with one of the vectors of the canonical base, that is, the true positives. Given these relations of inclusion, we derive the following relations on the sizes:

$$\|\mathbf{X}\| \leq \|\mathcal{C}_\varphi \cap \text{SPAN}(\mathbf{X})\| \leq \|\text{SPAN}(\mathbf{X})\|.$$

If we divide all terms by $\|\text{SPAN}(\mathbf{X})\| > 0$, we obtain:

$$\frac{\|\mathbf{X}\|}{\|\text{SPAN}(\mathbf{X})\|} \leq \frac{\|\mathcal{C}_\varphi \cap \text{SPAN}(\mathbf{X})\|}{\|\text{SPAN}(\mathbf{X})\|} \leq 1. \tag{2}$$

The number of vectors in $\mathbf{X}$ is, by construction, N. In a finite field, the size of $\text{SPAN}(\mathbf{X})$ is also a finite number, and can be computed as follow. Let $R = \text{RANK}(\mathbf{X}) \leq N$; by definition, there exist R linearly independent vectors $\widehat{\mathbf{x}}_1, \ldots, \widehat{\mathbf{x}}_R$ in $\mathbf{X}$. The span of $\mathbf{X}$ is the set of all vectors $\mathbf{z}$ such that:

$$\mathbf{z} = \sum_{i=1}^{R} \omega_i \widehat{\mathbf{x}}_i.$$

There are therefore R variables ω_i that can be freely chosen in $\mathbb{F}_q$ to combine the independent vectors of $\mathbf{X}$, leading to a total of q^R distinct combinations. By substitution in (2), we obtain:

$$\frac{N}{q^R} \leq P(\mathbf{X}) \leq 1.$$

Once we have found the relative number of positives for any given matrix $\mathbf{X}$, we can compute the expected relative number of positives for the whole code $\mathcal{C}_\varphi$ by averaging $P(\mathbf{X})$ over all $\mathbf{X}$s:

$$\begin{aligned} P(\mathcal{C}_\varphi) &= \text{E}\left[P(\mathcal{X})\right] \\ &= \sum_{\mathbf{X}} P(\mathbf{X}) \Pr\{\mathcal{X} = \mathbf{X}\} \\ &\geq \frac{N}{q^{\overline{R}}}, \end{aligned}$$

where $\overline{R} = \text{E}\left[\text{RANK}(\mathcal{X})\right]$.

Since the rank of any $\mathbf{X}$ can be at most N, we can say that for any choice of the function $\varphi(\cdot)$, thus for any code $\mathcal{C}_\varphi$, $P(\mathcal{C}_\varphi) \geq Nq^{-N}$.

Let us now see how the Practical Network Coding can be interpreted within our framework. In the case of PNC, the code $\mathcal{C}_\varphi$ is not constructed in advance: once a set of source messages $\mathbf{s}_1, \ldots, \mathbf{s}_N$ has been selected, the code is constructed in such way that $\varphi(\mathbf{s}_i) = \mathbf{e}_i$ (where $\mathbf{e}_i$ is the i-th vector of the canonical base of $\mathbb{F}_q^N$) for the source messages, and $\varphi(\mathbf{s}) = \mathbf{0}$ for any vector $\mathbf{s} \in \mathbb{F}_q^T \setminus \{\mathbf{s}_1, \ldots, \mathbf{s}_N\}$. Notice that is always possible to construct such a function $\varphi(\cdot)$ as we assumed that the sources are known and distinct, thus it is always possible to deduce the index i from the value of $\mathbf{s}_i$, and to associate the corresponding vector $\mathbf{e}_i$ of the base.

Let us consider the matrix $\mathbf{\Phi}$ defined as the collections of the vectors $\varphi(\mathbf{s}_i)$ for every source message $\mathbf{s}_i$:

$$\mathbf{\Phi} = \begin{pmatrix} \varphi(\mathbf{s}_1) \\ \vdots \\ \varphi(\mathbf{s}_N) \end{pmatrix} = \begin{pmatrix} \mathbf{e}_1 \\ \vdots \\ \mathbf{e}_N \end{pmatrix} = \mathbf{I}_N.$$

Since $\mathbf{\Phi}$ is the canonical base, its image is $\mathbb{F}_q^N$ and its kernel is $\{\mathbf{0}\}$; therefore, by definition, no non-zero linear combination of the vectors in $\mathbf{X}$, which is a concatenation of the original sources and their digests, can be a codeword of $\mathcal{C}_\varphi$ other than one of the vectors of $\mathbf{X}$ themselves, *i.e.*, $\mathcal{C} \cap \text{SPAN}(\mathbf{X}) = \mathbf{X}$, and $P(\mathcal{C}) = Nq^{-N}$ (and indeed, we find only the N source packets over the q^N possible linear combinations of the vectors in $\mathbf{X}$). It is worth noticing that albeit the PNC approach is equivalent to a perfectly non-linear function, it has one operating point only, in the sense that the length D of the digest $\varphi(\cdot)$ is fixed(namely $D = N$) and it cannot be generalized. Our framework, on the other hand, is more general and allows to design a system in which different values of D can be used, resulting in different values of $P(\mathcal{C})$.

Our results show that in order to minimize the number of false positives, the optimal hashing function $\varphi^*(\cdot)$ has to generate for the original sources $\mathbf{S}$ a set of digests $\mathbf{\Phi}$ such that its expected rank is maximized.

The correct message digest will thus point to a set of candidates for the original sources much smaller than the original search space, on which other criteria –like entropy minimization– can be applied.

Since the source matrix $\mathbf{S}$ is by hypothesis random, so will be the digest matrix $\mathbf{\Phi}$. Rank maximization for a random matrix in a finite field is a challenge commonly found in Practical Network Coding (PNC) [5] for the generation of the mixing matrix, and it is commonly solved by selecting the coefficients uniformly from the field.

We therefore need to design a hashing function that for a message of arbitrary length T can generate an assigned number D of digest symbols such that these symbols are uniformly distributed in the finite field.

For this purpose, we propose to use a *sponge construction* of the hashing function [25]. A sponge construction is a hashing function design technique that allows to decouple the input length and the output length of the hashing function, depicted in Fig. 3. This allows to generate an arbitrary length digest for inputs of any length. Two primitive functions are provided: first an ABSORB function that takes a variable-length input $\mathbf{S}$ and produces a fixed-length state Q, then a SQUEEZE function that takes the state Q and returns an output $\mathbf{\Phi}$ of arbitrary size D specified by the user.

Algorithm 1 Absorb part of the sponge construction of the hashing function. Given an input of arbitrary length $\mathbf{S}$, it produces a state Q of fixed length. The symbols $\oplus$ and $[\cdot]_{\text{r}}$ denote modulo-2 sum and circular right shift, respectively.

function Q =ABSORB($\mathbf{S}$)
  $\mathbf{S}$ is divided into L blocks B_i of 32 bits;
  $\sigma \leftarrow 0$; $K \leftarrow$ 0x99999999; $Q \leftarrow 0$;
  for $i \leftarrow 1$ to L **do**
    $Q \leftarrow [Q \oplus B_i]_{\text{r}}$;
    $Q \leftarrow [Q \oplus \sigma]_{\text{r}}$;
    $Q \leftarrow [Q \oplus K]_{\text{r}}$;
    $\sigma \leftarrow Q$;
    $Q \leftarrow Q \oplus [Q]_{\text{r}}$;
  end for
  return Q
end function

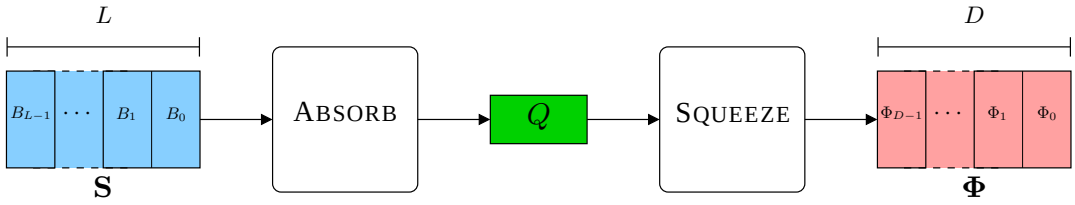


Figure 3. Sponge construction of the hashing function. The ABSORB function processes an input $\mathbf{S}$, divided in L blocks of fixed size, and produces a fixed-length state Q. The SQUEEZE function can use the state Q to generate a signature $\mathbf{\Phi}$ of assigned length D.

Algorithm 2 Squeeze part of the sponge construction of the hashing function. Given a state Q of fixed length it produces a message digest $\mathbf{\Phi}$ of assigned length D. The symbols $\oplus$ and $[\cdot]_{\mathrm{r}}$ denote modulo-2 sum and circular right shift, respectively.

function $\mathbf{\Phi}$ =SQUEEZE(Q, D)
  $\sigma \leftarrow 0$; $K \leftarrow$ 0x99999999;
  for $i \leftarrow 1$ to D **do**
    $Q \leftarrow [Q]_{\mathrm{r}}$;
    $Q \leftarrow [Q \oplus \sigma]_{\mathrm{r}}$;
    $Q \leftarrow [Q \oplus K]_{\mathrm{r}}$;
    $\sigma \leftarrow Q$;
    $Q \leftarrow Q \oplus [Q]_{\mathrm{r}}$;
    $\Phi_i \leftarrow Q$;
  end for
  return $\mathbf{\Phi}$
end function

In order to process a variable length input, the ABSORB function works on blocks of data of fixed length (in our implementation, 32 bits). The input data might need to be zero-padded to fit in an integer number L of blocks.

The implementations of the ABSORB and SQUEEZE functions are given in Algorithm 1 and 2, respectively. Note that the functions perform the same basic operations, with different inputs and outputs. These functions use basic bit operations commonly used in hashing: modulo-2 sum (*i.e.*, exclusive or) and circular right shift, denoted in Algorithm 1 and 2 by $\oplus$ and $[\cdot]_{\mathrm{r}}$, respectively. In the ABSORB function, the state Q is initialized to zero. Then, for each iteration i, one of the L blocks B_i of the input is added to the current state in modulo 2. The result is then circularly shifted by one position.

The same operations of update of the state –*i.e.*, sum and circular shift– are then applied using the value state at the previous iteration σ, and a constant value K. The constant value is chosen to prevent that a long run of zeros in the input might permanently force the state to zero. Finally the state is added to the shifted version of itself.

In the SQUEEZE function, the operations are the same, except that the blocks B_i are replaced with constant zero blocks, while the output $\mathbf{\Phi}$ is composed of the state Q at the end of each iteration i. The number D of iterations, equal to the number of output symbols, is specified by the user.

These functions work on blocks of fixed size of 32 bits, therefore, in order to produce outputs in fields smaller than $\mathbb{F}_{2^{32}}$, only the first b bits of each symbol Φ_i are considered. Notice that this computations are easy, and can be implemented efficiently at low level.

Although it would be extremely difficult to compute the expected number of positives for this hashing function, due to the finite nature of $\mathbb{F}_q^{T'}$ it is in principle possible to compute $P(\mathcal{C}_\varphi)$ by full exploration. However, a good assessment of the quality of the function can be provided –without the computational aggravation of a full exploration– by simply comparing its performances with the lower bound Nq^{-N} on a statistically sufficient number of matrices $\mathbf{X}$. Such a comparison is presented in Fig. 4 for $N = 4$ sources in finite fields $\mathbb{F}_2$, $\mathbb{F}_4$ and $\mathbb{F}_8$.

We observe that in all the scenarios our function is almost equivalent to the theoretical optimum. We notice that –as expected– for $D = N$ the performance of the hashing technique becomes extremely close to that of PNC; however, it is important to notice that PNC appears in this figures as a single point –reflecting the fact that its overhead is fixed for a given generation size– while the hashing technique allows a trade-off between overhead and expected number of a false positive. Notice that for $D = \frac{N}{2}$, *i.e.*, with half of the overhead w.r.t. network coding, the loss in performance is almost negligible. This same behavior has been observed also for $N = 2$, $N = 4$ and $N = 8$ (not shown here for the sake of brevity).

Finally, the separation procedure based on this hashing function is presented in Algorithm 3. For each vector $\mathbf{w}$ of length N in $\mathbb{F}_{2^b}$, we try to *demix* one message $\mathbf{z}$ and the respective digest $\mathbf{\Phi_z}$. If the digest is valid, *i.e.*, if $\varphi(\mathbf{z}) = \mathbf{\Phi_z}$, we store in $\mathcal{V}$ the combination vector $\mathbf{w}$. After all the vectors $\mathbf{w}$ have been tried, we select the N linearly independent vectors in $\mathcal{V}$ corresponding to the demixed messages with the lowest entropy. The matrix $\mathbf{W}$ composed as the horizontal concatenation of these vectors is our estimation of the inverse matrix of $\mathbf{A}$. We limit ourselves to a family of linearly independent vectors under the assumption that, being $\mathbf{W}$ the inverse of $\mathbf{A}$, it has full rank N. The demixed message corresponding to this matrix $\widehat{\mathbf{Z}}$ will represent our estimation of the encoded sources.

III. EXPERIMENTAL RESULTS

In the following, we present the results for the separation of N sources defined in a finite field $\mathbb{F}_{2^b}$, for the proposed digest-enhanced technique, and compare them with the results achievable using an exhaustive entropy-based technique without overhead, such as described in Sec I.

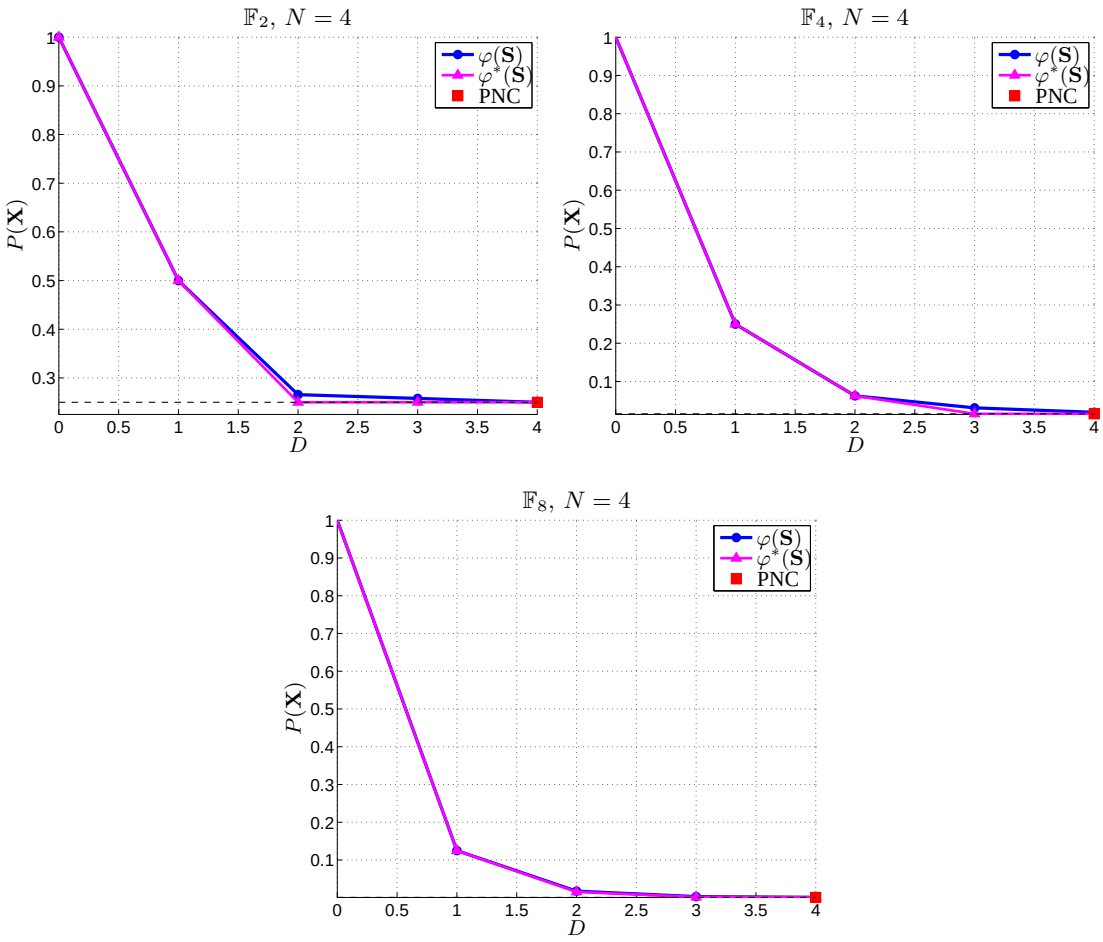
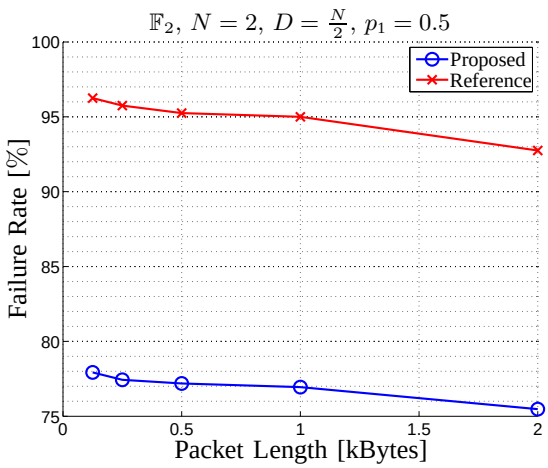
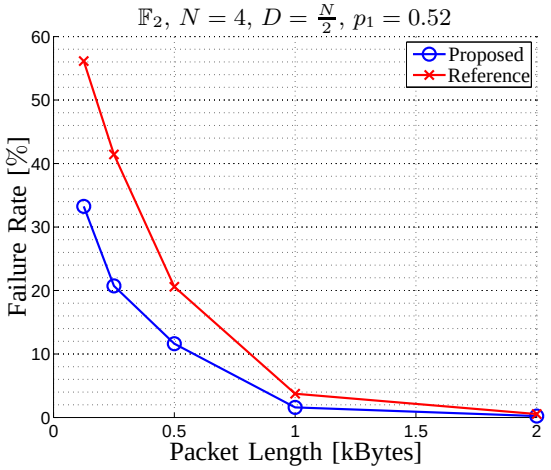


Figure 4. Comparison between our sponge-constructed hashing function $\varphi(\cdot)$, the theoretical optimal hashing function $\varphi^*(\cdot)$, and Practical Network Coding. The relative frequency positive $P(\mathbf{X}) = \Pr\{\mathbf{z} \in \mathcal{C}_\varphi \mid \mathbf{z} = \mathbf{w}^\top \mathbf{X}\}$ is plotted against the size of the overhead. The dashed black line represents the value $P(\mathbf{X}) = \frac{N}{q^N}$, corresponding to the case where only the sources are identified as codewords, *i.e.*, the case of perfect separation.



(a) Comparison for N=2 sources. The source bits are one with probability p_1=0.5. The overhead is 1 symbol per packet.



(b) Comparison for N=4. The source bits are one with probability p_1=0.52. The overhead is 2 symbols per packet.

Figure 5. Comparison between the reference entropy-based method and the proposed digest-enhanced technique, for in $\mathbb{F}_2$. The failure rate, *i.e.*, the percentage of sources that the algorithm was *not* able to identify, is plotted against the packet length (in kilobytes).

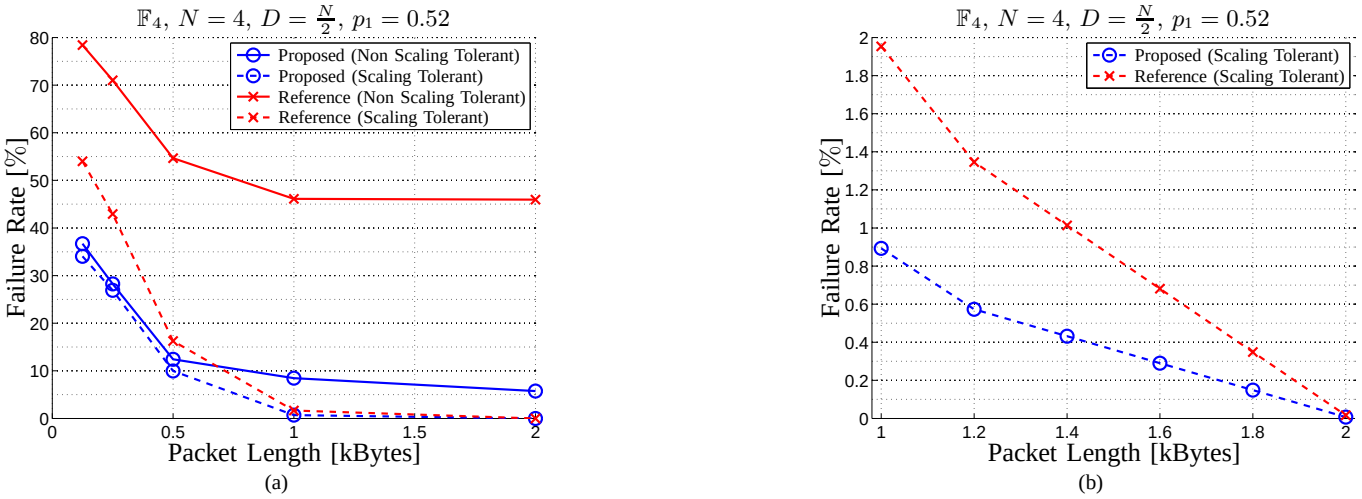


Figure 6. Comparison between the reference method and the proposed technique, for N=4 sources in $\mathbb{F}_4$. The source bits are one with probability p_1=0.52. The overhead is 2 symbols per packet. The failure rate is plotted against the packet length (in kilobytes). The dashed lines represent the failure rate when the sources are considered identified up to a scaling factor. The solid lines represent the failure rate when scaling ambiguity is not tolerated. The plot on the right represents the same comparison, between 1 and 2 kilobytes.

Algorithm 3 Separation algorithm. Entropy minimization is applied only on those candidate solutions that carry a valid message digest.

Input: $(N \times T')$ observation matrix $\mathbf{Y}$.
Output: $(N \times T)$ separated source matrix $\widehat{\mathbf{S}}$.
$\mathcal{V} \leftarrow \emptyset$, $\mathcal{W} \leftarrow \emptyset$;
for all $\mathbf{w}$ of length N in $\mathbb{F}_{2^b}$ **do**
  $(\mathbf{z} \mid \mathbf{\Phi_z}) \leftarrow \mathbf{w}^\top \mathbf{Y}$;
  if $\varphi(\mathbf{z}) = \mathbf{\Phi_z}$ **then**
    $\mathcal{V} \leftarrow \mathcal{V} \cup \{\mathbf{w}\}$;
  end if
end for
repeat
  $\mathbf{w}^* \leftarrow \arg\min_{\mathbf{w} \in \mathcal{V}} \{H(\mathbf{w}^\top \mathbf{Y})\}$;
  if $\mathbf{w}^* \notin \text{SPAN}(\mathcal{W})$ **then**
    $\mathcal{W} \leftarrow \mathcal{W} \cup \{\mathbf{w}^*\}$;
  end if
  $\mathcal{V} \leftarrow \mathcal{V} - \{\mathbf{w}^*\}$;
until $\|\mathcal{W}\| = N$
$\widehat{\mathbf{W}} \leftarrow$ matrix built from the row vectors in $\mathcal{W}$;
$\left(\widehat{\mathbf{Z}} \mid \widehat{\mathbf{\Phi}}\right) \leftarrow \widehat{\mathbf{W}}^\top \mathbf{Y}$;
$\widehat{\mathbf{S}} \leftarrow \widehat{\mathbf{Z}}$;

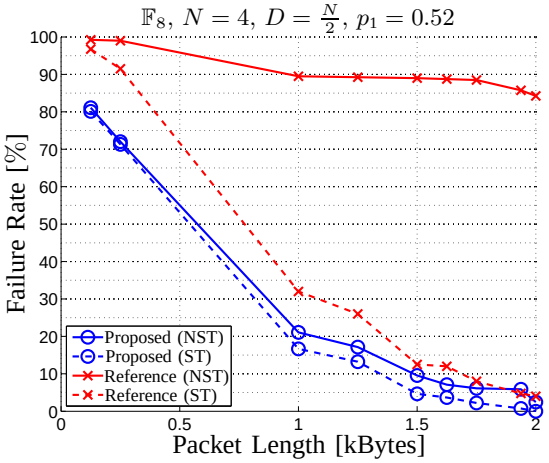


Figure 7. Comparison between the reference method and the proposed technique, for N=4 sources in $\mathbb{F}_8$. The source bits are one with probability p_1=0.52. The overhead is 2 symbols per packet. The failure rate is plotted against the packet length (in kilobytes). The dashed lines represent the failure rate when the sources are considered identified up to a scaling factor (*Scaling Tolerant*, ST), while the solid lines represent the failure rate when scaling ambiguity is not tolerated (*Non Scaling Tolerant*, NST).

In particular, in our experimental setup, the reference technique simply consists in identifying the N linear combinations of observations such that the combination coefficients are linearly independent and the entropy is minimized [20, 22]. This technique does not alter the sources and does not add any overhead.

Our technique, on the other hand, is restrained to the linear combinations of observations that carry a valid digest, *i.e.*, such that the digest appended to the packet is equal to the one locally computed by the separation algorithm.

In order to have a consistent parameter for comparison over different finite fields, the probability distributions of the sources are expressed in terms of p_1, *i.e.*, the probability that a bit is 1. For finite fields larger than $\mathbb{F}_2$, this probability is applied independently on each bit.

We report in Fig. 5(a) the *failure rate* of the technique *vs.* the length of the packet for the case of $N = 2$ sources in $\mathbb{F}_2$ with an overhead $D = \frac{N}{2} = 1$ symbol and with uniform distribution of the sources (the bit probability $p_1 = 0.5$). As we showed in Sec. II, an overhead of $D = \frac{N}{2}$, with the proposed hashing function, offers an excellent compromise between overhead and false positive probability. Notice that, in any scenario, PNC achieves a 100 % success rate (assuming a non-singular encoding matrix), at the cost of N symbols of overhead, for a generation of size N (twice as much than our technique). The packet length includes, for the proposed technique, the overhead —which is in any case of a few bits over several hundreds of bytes and therefore does not affect the

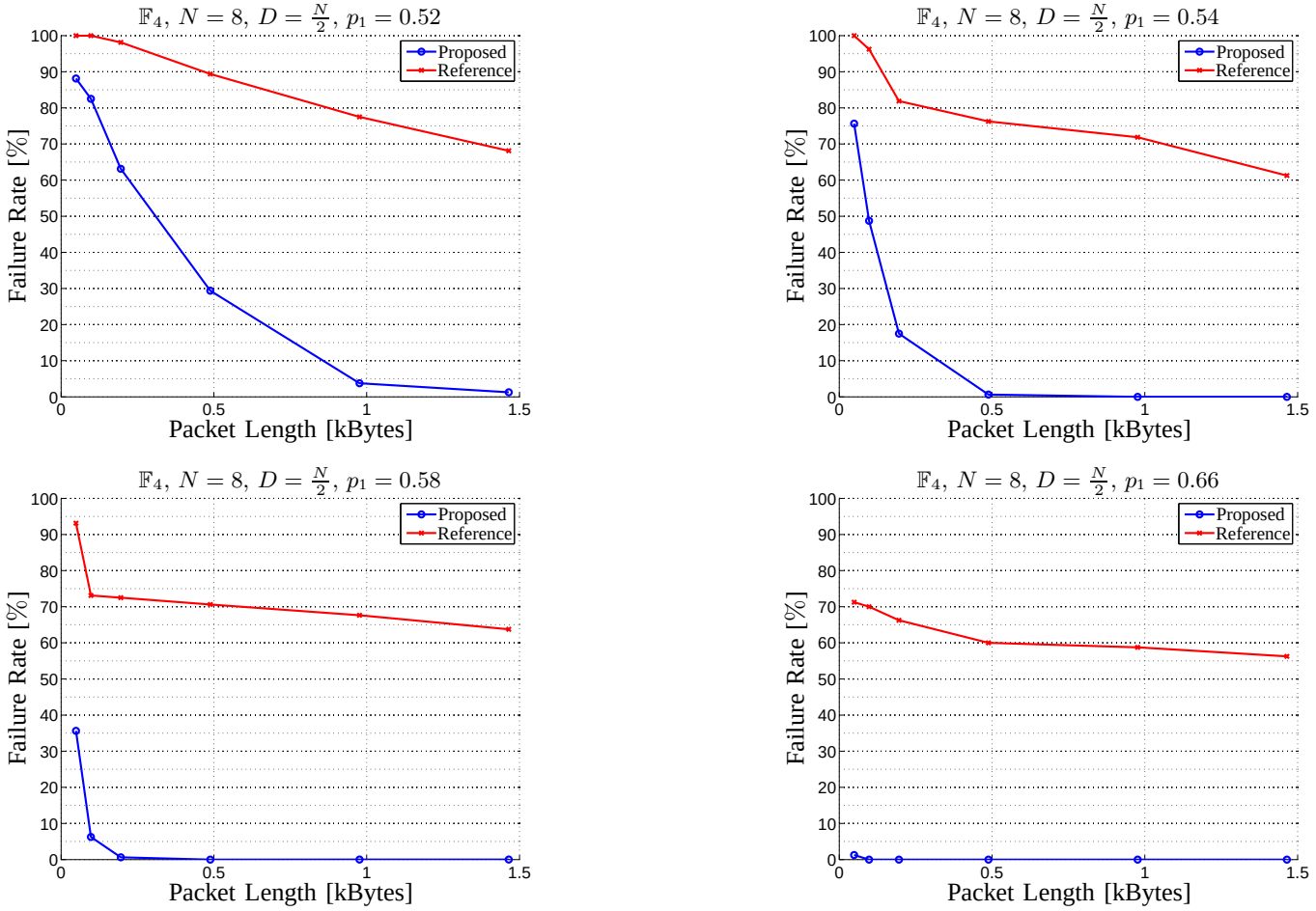


Figure 8. Comparison between the reference method and the proposed technique, for N=8 sources in $\mathbb{F}_4$, for source bits are one with probability p_1=0.52, 0.54, 0.56, 0.58, 0.66, and 0.75. The overhead is 4 symbols per packet. The failure rate is plotted against the packet length (in kilobytes).

figure. Each plotted point corresponds to the average over at least 100 runs of the algorithm, each with randomly generated sources and mixing matrix. The failure rate is simply one minus the success rate, where the success rate is the number of correctly identified sources divided by the total number of sources.

We observe that in the case where the ICA methods have the worst performance (*i.e.*, uniform distribution), even with just one bit of overhead per packet, our technique consistently outperforms the reference method, and the separation is greatly improved for all packet lengths. Notice that when the sources are uniform, entropy minimization methods are no better than a blind (*i.e.*, random) choice of the demixing coefficients, which means that our technique is able to identify up to 25 % of the sources by relying on the one-bit signature alone.

In Fig. 5(b), we also report the results obtained for a higher number of sources ($N = 4$) in the same field $\mathbb{F}_2$, again with an overhead $D = \frac{N}{2}$ symbols, in this case two bits. The most relevant difference is that, in this case, the sources are not uniform ($p_1 = 0.52$). This allows both methods to converge to complete separation with the length of the packets. However, our technique still consistently outperforms the reference, an effect more noticeable when the length of the packets is small and the failure rate is reduced by almost a factor two. This result is very important for practical applications, in which the separation is done packet-wise, since packets typically have a size limit dictated by the network.

Notice that, in practical applications, finite fields of order higher than two are typically used, as the probability of randomly generating a mixing matrix that has full rank –and is therefore invertible– increases with the size of the field.

In this respect, we present in Fig. 6(a) and 7 the results obtained if we consider the same scenario in terms of number of sources and source distribution, but with sources defined in $\mathbb{F}_4$ and $\mathbb{F}_8$, respectively.

As mentioned in Sec. I, entropy-based methods can only identify sources up to a scaling factor, a limitation known as *scaling ambiguity*. If we tolerate the scaling ambiguity, we observe that the performances for both methods are similar to the previous case. In particular, for $\mathbb{F}_4$ we observe a failure rate of about 55 % for the reference technique and 35 % for the proposed for packets of about 128 bytes, and a failure rate of less than 1 % for both techniques at about 1 kilobyte. Notice that, by using a more stringent definition of success, the non scaling tolerant failure rate is necessarily larger or equal than the scaling tolerant.

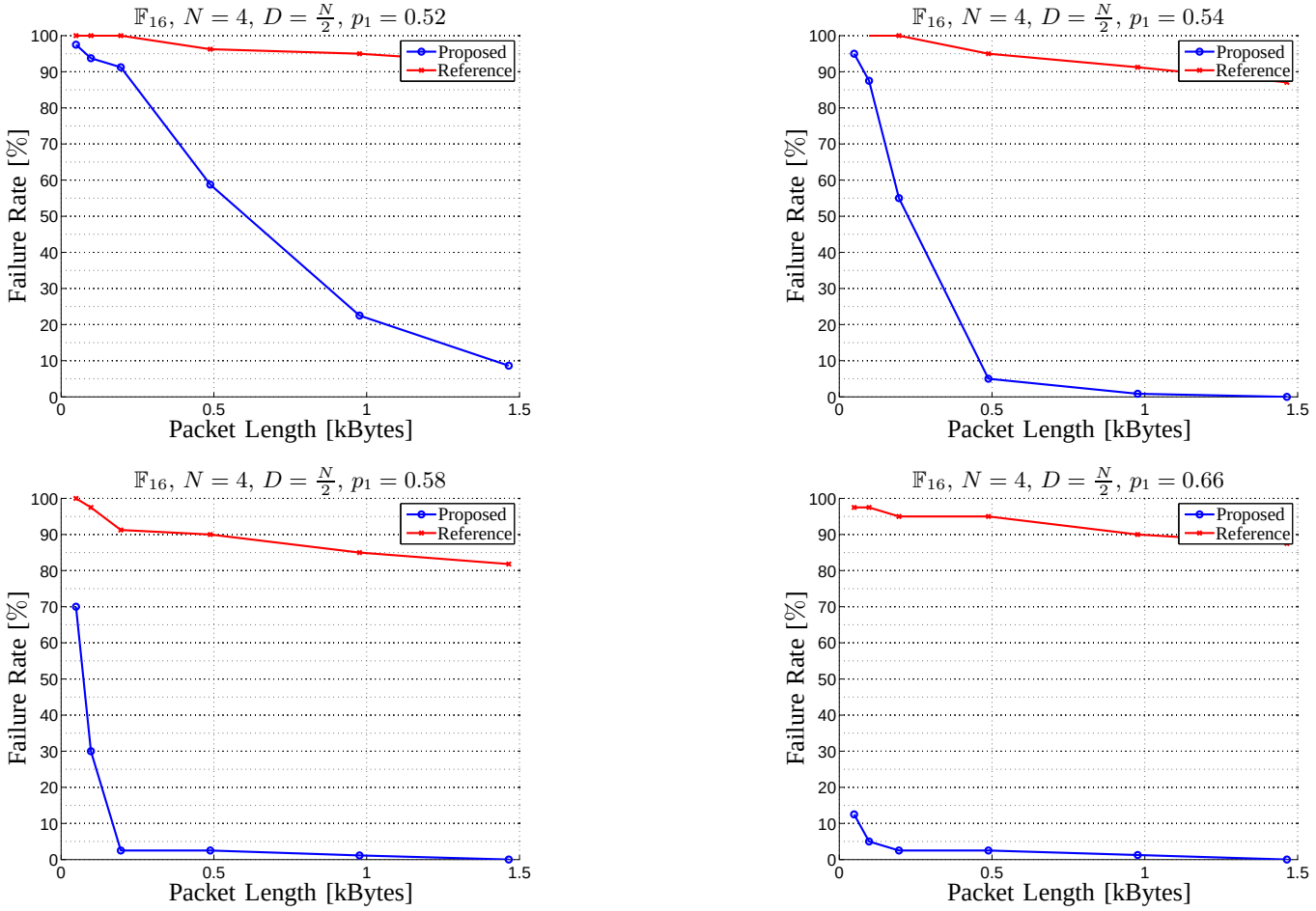


Figure 9. Comparison between the reference method and the proposed technique, for $N{=}4$ sources in $\mathbb{F}_{16}$, for source bits are one with probability $p_1{=}0.52$, 0.54, 0.56, 0.58, 0.66, and 0.75. The overhead is 4 symbols per packet. The failure rate is plotted against the packet length (in kilobytes).

However, unlike the case of analog applications, scaling ambiguity is often not tolerable in digital applications, *e.g.*, a multiple in finite field of an encoded video packet bares no meaning, and the scaled signal is not semantically equivalent to the unscaled one.

Therefore, if we consider that failure rate in a stricter sense, where we do not tolerate the scaling ambiguity, we see that our technique presents a much lower failure rate than the reference even for longer packets. In fact, without scaling ambiguity, the failure rate of the reference technique increases to $80,\%$ for packets of about 128 bytes, while it remains almost unaltered for the proposed. For sources of about 1 kilobyte, the failure rate of the reference technique is about $50\,\%$, while it stays lower than $10\,\%$ for the proposed one. Furthermore, while the failure rate of the proposed technique keeps decreasing when the length of the packets increases up to 2 kilobytes, the reference technique stays almost flat at $55\,\%$.

There may exist anyway, even in finite fields, applications that are tolerant to scaling ambiguity. For this reason, in Fig. 6(b) we also report a magnification of Fig. 6(a) in the range of packet lengths $1 \sim 2$ kilobytes. We observe that the failure rate of our technique is consistently lower than the one achieved by the reference technique, with a reduction of approximatively a factor two.

For the sake of completeness, in Figures 8, 9, and 10, we report the results for $N{=}8$ sources in $\mathbb{F}_4$, $N{=}4$ sources in $\mathbb{F}_{16}$, and $N{=}2$ sources in $\mathbb{F}_{256}$ respectively, each for source bits are one with probability $p_1{=}0.52$, 0.54, 0.56, 0.58, 0.66, and 0.75.

Furthermore, in Fig. 11, we compare our proposed technique with the alternative network coding overhead compression methods proposed by Thomos *et al.* [15] and discussed in Sec. I. In this scenario, we consider a generation of $N = 2$ sources, so that for both techniques, the overhead is 1 symbols per packet. We show the failure rate as a function of the logarithm of the size of the finite field. It should be noted that what is relevant in this comparison is the probability of correctly decoding the sources given a non-singular matrix, and the probability of having a non-singular encoding matrix itself. While the alternative method does have an almost $100\,\%$ value for the former, the latter is typically much lower. The combined effect is that, while our technique performs much better for small finite fields, the two techniques become closer around field $\mathbb{F}_{32}$, while the alternative method performs better for larger fields. It should be noted, however, that while the alternative method provides a viable way to reduce the

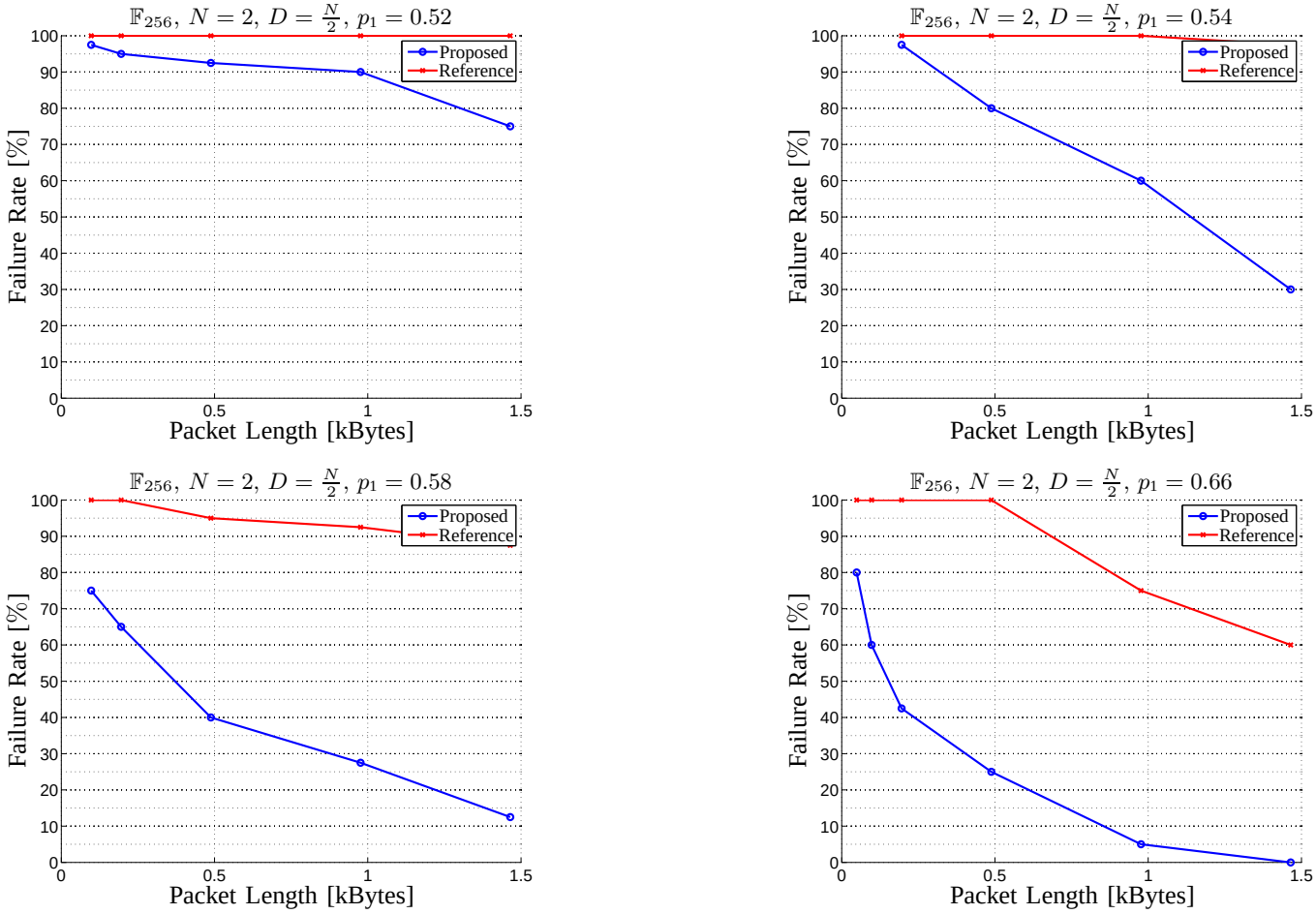


Figure 10. Comparison between the reference method and the proposed technique, for $N{=}2$ sources in $\mathbb{F}_{256}$, for source bits are one with probability $p_1{=}0.52$, 0.54, 0.56, 0.58, 0.66, and 0.75. The overhead is 1 symbols per packet. The failure rate is plotted against the packet length (in kilobytes).

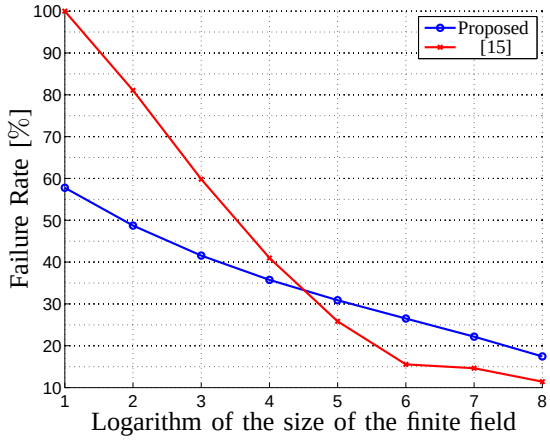


Figure 11. Comparison between the proposed technique and the alternative network coding overhead compression methods [15] for $N{=}2$ sources, whose bits are one with probability $p_1{=}0.52$. For both techniques, the overhead is 1 symbols per packet. The failure rate is plotted against the logarithm of the size of the finite field.

overhead, it still require that the nodes of the network reach a consensus on the ordering of the sources.

So far, we have presented the results of our proposed technique in separating signals that have been generated to have a given probability distribution. In order to validate our approach in a more realistic multimedia transmission scenario, in Figures 12, 13 and 14 we presents the results relative to the separation of image (JPEG), audio (MP3), and video (H.264/AVC) content. Notice that, in this case, we do not have control over the probability distribution of the data.

In all the scenarios, our technique consistently outperforms the reference. It should be noticed that, as the size of the finite field increases, the inherently difficulty of separation in a larger field is partially compensated by the fact that the higher-order entropy (*i.e.*, the entropy considering blocks of an increasing number of bits) becoming smaller. In summary, we observe that, in most of the considered scenarios, our technique provides a viable trade-off between decoding probability and overhead, suitable to be integrated in a unequal loss protection scheme beside traditional practical network coding. In the remaining cases, where the proposed technique alone is unable to provide an acceptable success rate for the application, it is still possible to exploit its advantages for a large fraction, while the remaining data (e.g. 25 % in Fig. 12 for $\mathbb{F}_{256}$) can be retransmitted, possibly using traditional network coding.

Finally, some considerations about the time-complexity of

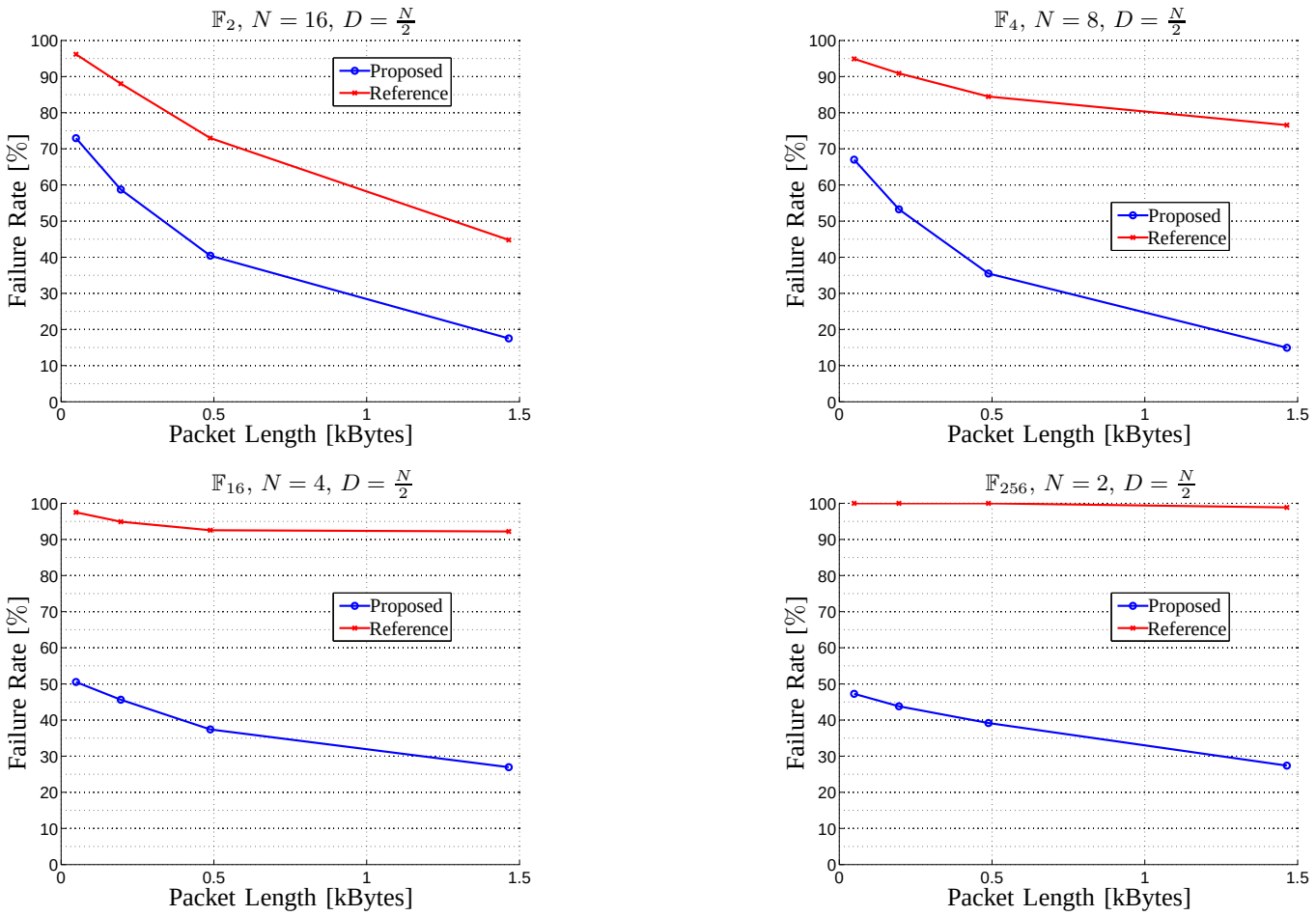


Figure 12. Comparison between the reference method and the proposed technique for transmission of image data (JPEG). The results are presented for different numbers of sources and different fields. The failure rate is plotted against the packet length (in kilobytes).

our proposed technique. As detailed in Sec. II, our techniques adds validity check, by means of digest validation, to an entropy minimization algorithm. This is done for each vector whose entropy would be measured in purely entropy-based technique, thus, its complexity depends on the complexity of the entropy minimization algorithm that is used. In this work, we choose to use the AMERICA algorithm, because of its simplicity and stability, so the time-complexity of the technique is $O\left(Tq^N\right)$, where T is the length of the packet, q is the size of the finite field, and N is generation size. A smaller complexity can be achieved if, instead of AMERICA, another ICA techniques, such as those mentioned in Sec. I, are used.

IV. Conclusions & Future Work

In this paper, we presented a novel approach to blind separation of source signals defined over a finite field. Building on our previous work, in which we proved that traditional entropy-based separation algorithms can be greatly improved if assisted with a non-linear error-detecting encoding of the sources, we proposed to generate, for each source, a non-linear message digest to be sent along the sources. The message digest is generated by a hashing function defined through a sponge-construction, which allows to decouple the input and the output length. In other words, the function is able to generate a digest of any given length for sources of arbitrary length. The message digest is defined to be robust w.r.t. linear combination, *i.e.*, a linear combination of digests has very low probability of being equal to the digest of the linear combination of the corresponding messages.

This property is exploited at the receiver side where observations with an invalid digest can be discarded without further processing. On the remaining observations, which are a considerable smaller subset of the search space, traditional entropy-based methods can be applied.

Our results show that this approach dramatically improves the separation ability of the technique, in cases where the traditional approaches are unfeasible, *i.e.*, for short sources with distributions close to uniform.

Furthermore, our technique is much more robust to the scaling ambiguity problem, which we argue is much more problematic in digital multimedia applications than it is in traditional analog blind source separation.

The possibility of separating efficiently the mixed sources given a small and controllable overhead open the possibility for a lossy network coding transmission scheme, where sources are linearly combined in order to increase throughput and loss immunity, but the overhead is significantly reduced

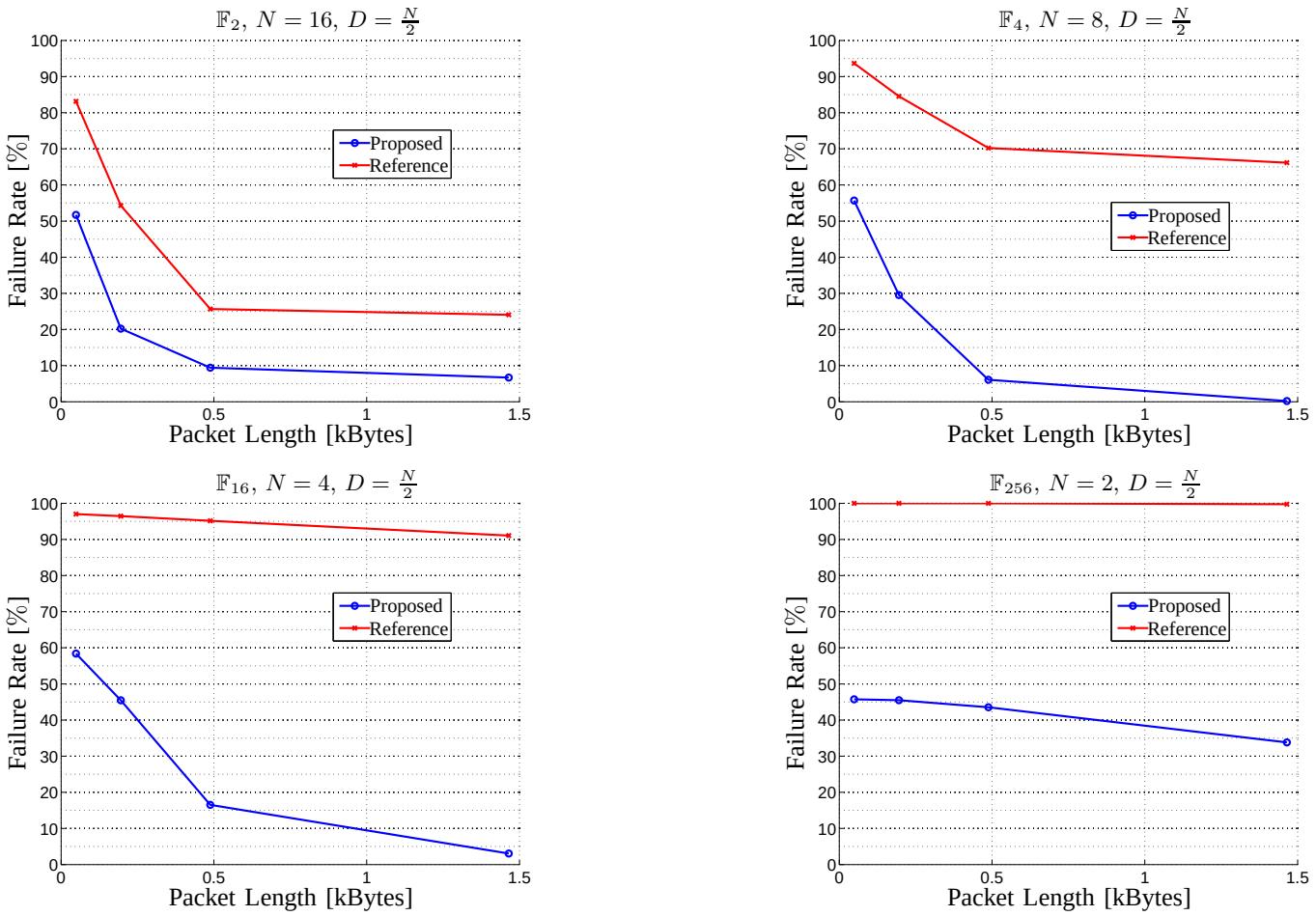


Figure 13. Comparison between the reference method and the proposed technique for transmission of audio data (MP3). The results are presented for different numbers of sources and different fields. The failure rate is plotted against the packet length (in kilobytes).

at the cost of a reduced decoding probability.

Our technique therefore allows to strike a trade-off between the decoding probability and the rate needed for the transmission. A possible evolution is to work out its integration within a rate-distortion framework as a form of unequal error protection: one could design a mechanism by which, while generations of packets with a large impact on the overall distortion are sent with a traditional network coding scheme, less important generations (*e.g.*, refinement level in a scalable technique) can be sent using the proposed technique, with an overhead proportional to their impact on the distortion.

The improvement of the performance of our technique with the non-uniformity of the sources suggests that an interesting perspective is its application to the case of highly non-uniform data, such as sparse signals (*e.g.*, in the case of compress sensing).

Acknowledgments

The authors wish to thank Marc Castella at the CITI department of Télécom & Management Sud-Paris for his valuable comments and the enriching discussions on the topic of error-detecting codes applied to source separation in finite fields.

References

[1] R. Ahlswede, N. Cai, S.-Y. Li, and R. Yeung, "Network information flow," *IEEE Transactions on Information Theory*, vol. 46, no. 4, pp. 1204–1216, Jul. 2000.

[2] S.-Y. R. Li, R. W. Yeung, and N. Cai, "Linear network coding," *IEEE Transactions on Information Theory*, vol. 49, no. 2, pp. 371–381, Feb. 2003.

[3] R. Koetter and M. Médard, "An algebraic approach to network coding," *IEEE/ACM Transactions on Networking*, vol. 11, no. 5, pp. 782–795, Oct. 2003.

[4] T. Ho, M. Médard, J. Shi, M. Effros, and D. Karger, "On randomized network coding," in *Proceedings of IEEE International Symposium on Information Theory*, Kanagawa, Japan, Jun. 2003.

[5] P. Chou, Y. Wu, and K. Jain, "Practical network coding," in *Proceedings of Allerton Conference on Communication Control and Computing*, Monticello, IL, USA, Oct. 2003.

[6] D. Vukobratović and V. Stanković, "Unequal error protection random linear coding for multimedia communications," in *Proceedings of IEEE Workshop on Multimedia Signal Processing*, Saint-Malo, France, Oct. 2010.

[7] N. Thomos and P. Frossard, "Network coding of rateless video in streaming overlays," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 20, no. 12, pp. 1834–1847, Dec. 2010.

[8] N. Thomos, J. Chakareski, and P. Frossard, "Prioritized distributed video delivery with randomized network coding," *IEEE Transactions on Multimedia*, vol. 13, no. 4, pp. 776–787, Aug.

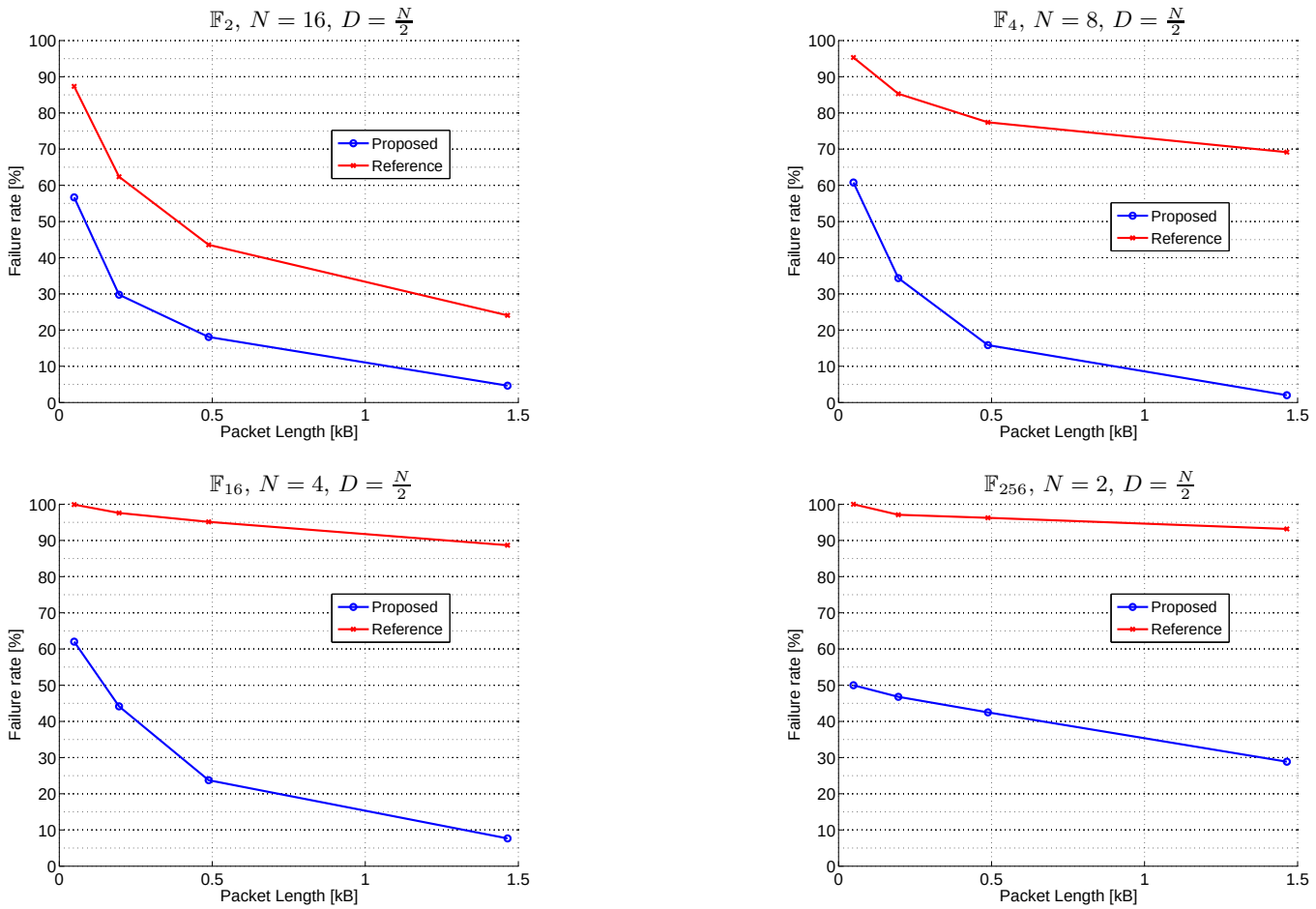


Figure 14. Comparison between the reference method and the proposed technique for transmission of video data (H.264/AVC). The results are presented for different numbers of sources and different fields. The failure rate is plotted against the packet length (in kilobytes).

2011.

[9] I. D. Nemoianu, C. Greco, M. Cagnazzo, and B. Pesquet-Popescu, "A framework for joint multiple description coding and network coding over wireless ad-hoc networks," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, Kyoto, Japan, Mar. 2012.

[10] C. Greco, I. D. Nemoianu, M. Cagnazzo, and B. Pesquet-Popescu, "A network coding scheduling for multiple description video streaming over wireless networks," in *Proceedings of European Signal Processing Conference*, Bucharest, Romania, Aug. 2012.

[11] D. Vukobratovic and V. Stankovic, "Unequal error protection random linear coding strategies for erasure channels," *IEEE Transactions on Communications*, vol. 60, no. 5, pp. 1243 – 1252, May 2012.

[12] E. Magli, M. Wang, P. Frossard, and A. Markopoulou, "Network Coding meets multimedia: A review," *IEEE Transactions on Multimedia*, Oct. 2012, accepted for publication.

[13] M. Jafari, L. Keller, C. Fragouli, and K. Argyraki, "Compressed network coding vectors," in *Proceedings of IEEE International Symposium on Information Theory*, Seoul, Republic of Korea, Jun. 2009, pp. 109–113.

[14] S. Li and A. Ramamoorthy, "Improved compression of network coding vectors using erasure decoding and list decoding," *IEEE Communications Letters*, vol. 14, no. 8, pp. 749–751, 2010.

[15] N. Thomos and P. Frossard, "Toward one symbol network coding vectors," *IEEE Communications Letters*, vol. 16, no. 11, pp. 1860–1863, 2012.

[16] P. Comon and C. Jutten, *Handbook of Blind Source Separation: Independent Component Analysis and Applications*, 1st ed. Academic Press, 2010.

[17] J.-F. Cardoso, "Blind signal separation: statistical principles," *Proceedings of the IEEE*, vol. 86, no. 10, pp. 2009–2025, 1998.

[18] P. Comon, "Independent component analysis, a new concept?" *Signal Processing (Elsevier Science)*, vol. 36, no. 3, pp. 287–314, Apr. 1994. [Online]. Available: http://dx.doi.org/10.1016/0165-1684(94)90029-9

[19] A. Hyvärinen and E. Oja, "Independent component analysis: algorithms and applications," *Elsevier Journal on Neural Networks*, vol. 13, no. 4-5, pp. 411–430, May 2000. [Online]. Available: http://dx.doi.org/10.1016/S0893-6080(00)00026-5

[20] A. Yeredor, "ICA in Boolean XOR mixtures," in *Proceedings of Springer-Verlag Internantional Conference on Independent Component Analysis*, 2007, pp. 827–835.

[21] H. Gutch, P. Gruber, and F. Theis, "ICA over finite fields," in *Proceedings of Springer-Verlag International Conference on Latent Variable Analysis and Signal Separation*, 2010, pp. 645–652.

[22] H. W. Gutch, P. Gruber, A. Yeredor, and F. J. Theis, "ICA over finite field–Separability and algorithms," *Signal Processing (Elsevier Science)*, vol. 92, no. 8, pp. 1796–1808, 2012.

[23] A. Yeredor, "Independent component analysis over Galois Fields of prime order," *IEEE Transactions on Information Theory*, vol. 57, no. 8, pp. 5342–5359, Aug. 2011.

[24] I. D. Nemoianu, C. Greco, M. Cagnazzo, and B. Pesquet-Popescu, "On a practical approach to source separation over

finite fields for network coding applications," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, Vancouver, BC, Canada, May 2013.

[25] G. Bertoni, J. Daemen, M. Peeters, and G. Van Assche, "The Keccak SHA-3 submission," *Submission to NIST (Round 3)*, 2011.

Irina Delia Nemoianu Irina Delia Nemoianu (S'11) received her engineering degree in Electronics Telecommunications and Information Technology in 2009, from the "Politehnica" University, Bucharest, Romania and her PhD degree in Signal and Image Processing in 2013, from Télécom ParisTech, France. Her research interests include advanced video service, wireless networking, network coding, and source separation in finite fields.

Claudio Greco Claudio Greco (M'13) received his his laurea magistrale in Computing Engineering (with honors), equivalent to an M.Sc., from the Federico II University of Naples, Italy in 2007, and his Ph.D. in Signal and Image Processing in 2012, from Télécom ParisTech, France, defending a doctoral thesis on robust broadcast of real-time video over wireless network. He is currently post-doctoral fellow at INRIA Rocquencourt on a shared project with Telecom-ParisTech and the L2S research unit. His research interests include multiple description video coding, multi-view video coding, mobile ad-hoc networking, cooperative multimedia streaming, cross-layer optimization for multimedia communications, and network coding.

Marco Cagnazzo (M'05-SM'11) obtained the Laurea (equivalent to the M.S.) degree in Telecommunication Engineering from Federico II University, Napoli, Italy, in 2002, and the Ph.D. degree in Information and Communication Technology from Federico II University and the University of Nice-Sophia Antipolis, Nice, France in 2005.

He was a post-doc fellow at I3S Laboratory (Sophia Antipolis, France) from 2006 to 2008. Since February 2008 he has been Associate Professor at Institut Mines-Télécom, Télécom ParisTech (Paris), within the Multimedia team. He is author of more than 90 contributions in peer-reviewed journals, conferences proceedings, books and book chapters. His current research interests are three-dimensional video communication and coding, distribued video coding, robust video delivery, network coding.

Dr. Cagnazzo is an Area Editor for *Elsevier Signal Processing: Image Communication* and *Elsevier Signal Processing*. Moreover he is a reviewer for major international scientific reviews (IEEE TRANS. MULTIMEDIA, IEEE TRANS. IMAGE PROCESSING, IEEE TRANS. SIGNAL PROCESSING, IEEE TRANS. CIRC. SYST. VIDEO TECH., *Elsevier Signal Processing, Elsevier Sig. Proc. Image Comm.*, and others) and conferences (IEEE International Conference on Image Processing, IEEE MMSP, European Signal Processing Conference, and others).

Béatrice Pesquet-Popescu Béatrice Pesquet-Popescu (SM'06, F'13) received the engineering degree in telecommunications from the "Politehnica" Institute in Bucharest in 1995 (highest honours) and the Ph.D. degree from the Ecole Normale Supérieure de Cachan in 1998. In 1998, she was a Research and Teaching Assistant with Université Paris XI, Paris. In 1999, she joined Philips Research France, Suresnes, France, where she worked for two years as a Research Scientist, then as a Project Leader, in scalable video coding. Since Oct. 2000 she is with Télécom ParisTech (formerly, ENST), first as an Associate Professor, and since 2007 as a Professor, Head of the Multimedia Group. She is the Head of the UBIMEDIA common research laboratory between Alcatel-Lucent and Institut Télécom. Her current research interests are in source coding, scalable, robust and distributed video compression and sparse representations. Dr. Pesquet-Popescu was an EURASIP BoG member (2003-2010), and an IEEE Signal Processing Society IVMSP TC member and MMSP TC associate member. She serves as an Associate Editor for IEEE Trans. on Image Processing, IEEE Trans. on Multimedia, IEEE Trans. on CSVT, Elsevier Image Communication, and Hindawi Int. J. Digital Multimedia Broadcasting journals and was till 2010 an Associate Editor for Elsevier Signal Processing. She was a Technical Co-Chair for the PCS2004 conference, and General Co-Chair for IEEE SPS MMSP2010, EUSIPCO 2012, and IEEE SPS ICIP 2014 conferences. Beatrice Pesquet-Popescu is a recipient of the "Best Student Paper Award" in the IEEE Signal Processing Workshop on Higher-Order Statistics in 1997, of the Bronze Inventor Medal from Philips Research and in 1998 she received a "Young Investigator Award" granted by the French Physical Society. She holds 23 patents in wavelet-based video coding and has authored more than 290 book chapters, journal and conference papers in the field. In 2006, she was the recipient, together with D. Turaga and M. van der Schaar, of the IEEE Trans. on Circuits and Systems for Video Technology "Best Paper Award".